歩行パターン生成

中島 崇晴

目次

- はじめに
- ２次元上での歩行パターン生成
- ３次元上での歩行パターン生成
  - 歩行素片
  - 歩行パラメータ
  - 着地点調整によるパターン生成
  - 方向転換
- 両足支持機関の導入
- 最後に
- 引用元

はじめに

- 【範囲】ヒューマノイドロボット
  - 4.2.5　簡単な2足歩行パターンの計画
  - 4.3.3　3次元の歩行パターン生成
  - 4.3.4　両足支持期間の導入

- 定義
  - sinθ = sθ
  - cosθ= cθ

2次元上での歩行パターン生成

2次元上での歩行パターン生成

なぜ、歩行パターンを生成するのか？

- 適切な拘束平面を設定することで、同じパターンを階段や不整地なに適応することができるから

2次元上での歩行パターン生成

- 簡単な歩行軌道を設計
  - 下の図のように理想的な歩行ロボットが平面上を一歩だけ進んで停止するものとする

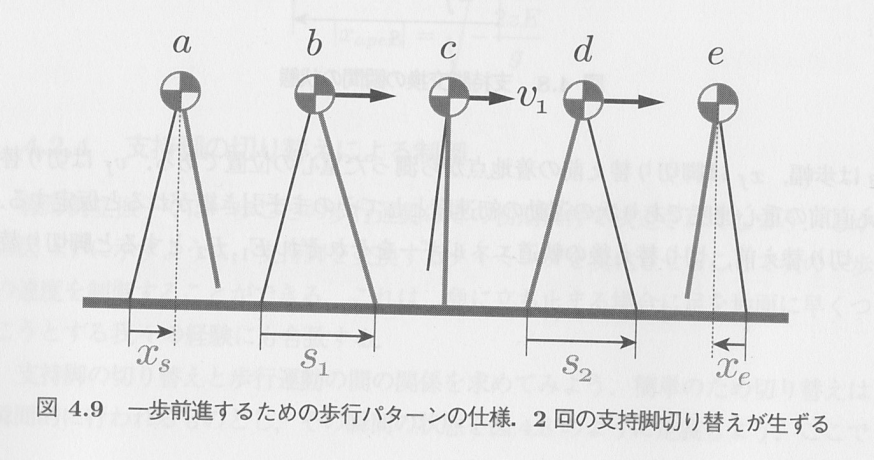


図 4.9　一歩前進するための歩行パターンの仕様．2 回の支持脚切り替えが生ずる

２次元上での歩行パターン生成

- 歩行開始時の運動（a→ｂ）

$$E_0 = -\frac{g}{2z}x_s^2$$

- 最初の支持脚切り替えから次の切り替えの間の運動（b →c →d）

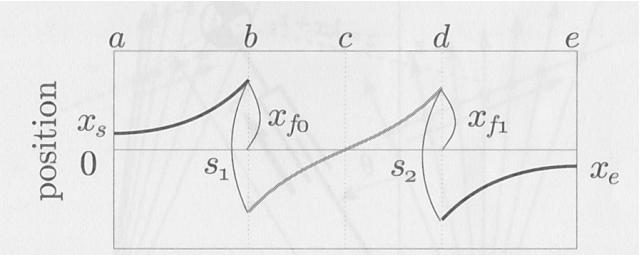
$$E_1 = \frac{1}{2}v_1^2$$

- 歩行終了時の運動（ｄ→e）

$$E_2 = -\frac{g}{2z}x_e^2$$

２次元上での歩行パターン生成

- 右下の式を使う
  - E0,E1から最初の支持脚交換条件を xf0
  - E1,E2から２番目の支持脚交換条件を xf1



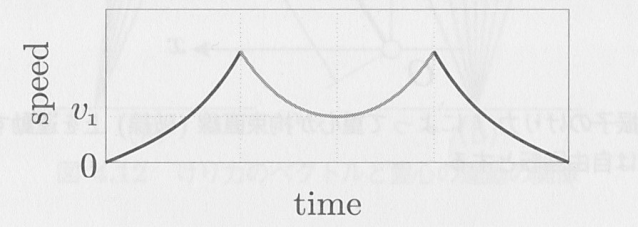
- 希望する歩行運動はこれらの条件で支持脚切り替えが起こるように遊脚を制御できる

$$x_f = \frac{z}{gs}(E_2 - E_1) + \frac{s}{2}$$

2次元上での歩行パターン生成

- 下の式により名切り替え時の速度も得られるので、一歩ごとの軌道が決定される

$$v_f = \sqrt{2E_1 + \frac{g}{z}x_f^2}$$



3次元上での歩行パターン生成

3次元上での歩行パターン生成

- 右の図は3次元線形倒立振子に基づく平面上の歩行パターンの例

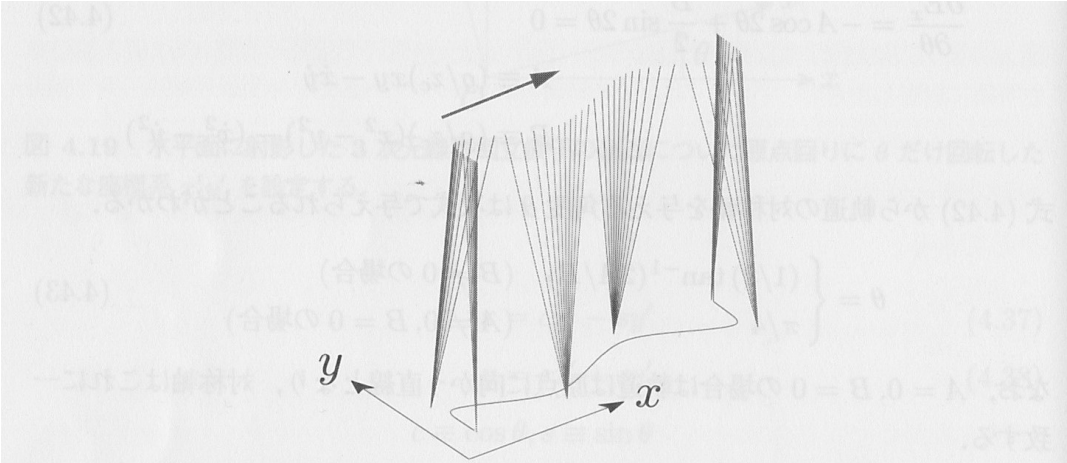
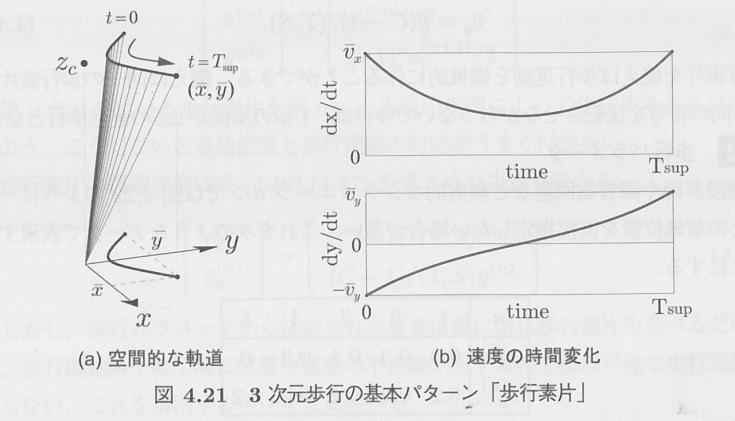


図 4.20　3 次元線形倒立振子による平面上の直進歩行パターン．静止状態から前へ 3 歩歩いて止まる

3次元上での歩行パターン生成

- 歩行素片
  - 運動方程式を用いて、ロボットが片足で支持している間の重心の軌跡を表したパターンを歩行素片という。
    - ヒューマノイドロボットのフィードバック制御より



(a) 空間的な軌道　　(b) 速度の時間変化

図 4.21　3 次元歩行の基本パターン「歩行素片」

3次元上での歩行パターン生成

- 歩行パラメータ
  - 階段や障害物回避などの現実的なシミュレーションでは
    - 一歩ごとの位置調整をしたい

  - 右の図のようなデータで表現する

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.3	0.3	0.3	0
$s_y^{(n)}$	0.2	0.2	0.2	0.2	0.2

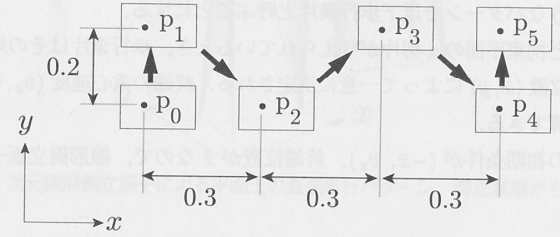


図 4.22　着地位置の設定：1 歩ごとの着地点 $p_0 \ldots p_N$ を設定し，ここから歩幅を決定する．点を囲む長方形はロボットの足跡のつもりである

  - sxが前進方向の歩幅
  - syが左右方向の歩幅

3次元上での歩行パターン生成

- 歩行パラメータ
  - 第n歩目の着地位置（px, py）

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} + s_x^{(n)} \\ p_y^{(n-1)} - (-1)^n s_y^{(n)} \end{bmatrix}$$

  - 第n歩目で用いるべき歩行素片のパラメータ（x，y）

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

  - 歩行素片の終端速度(vx, vy)

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} (C+1)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix}$$

3次元上での歩行パターン生成

- 着地点調整によるパターン生成
  - 着地のタイミングや周期を予め決められている歩行

- 一歩ごとの着地点を調節することで歩行速度を変化できる

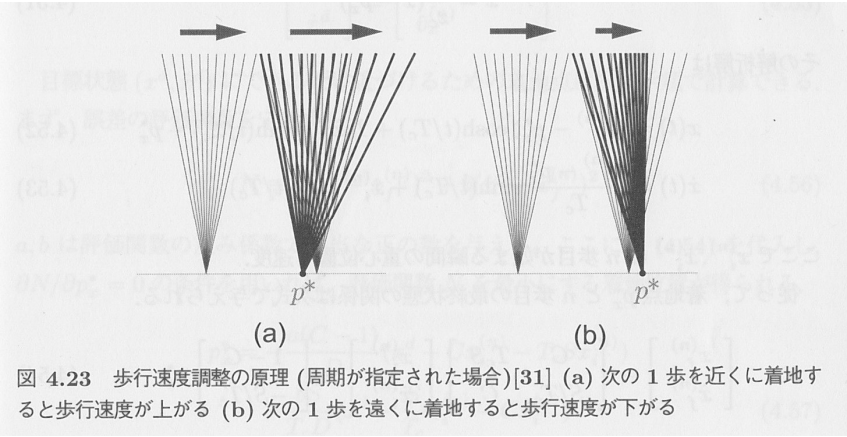


図 4.23　歩行速度調整の原理 (周期が指定された場合)[31] (a) 次の 1 歩を近くに着地すると歩行速度が上がる (b) 次の 1 歩を遠くに着地すると歩行速度が下がる

3次元上での歩行パターン生成

- 着地点調整によるパターン生成
  - 着地点px*とn歩目の最終状態の関係は下の式で与えられる

$$\begin{bmatrix} x_f^{(n)} \\ \dot{x}_f^{(n)} \end{bmatrix} = \begin{bmatrix} C & T_c S \\ S/T_c & C \end{bmatrix} \begin{bmatrix} x_i^{(n)} \\ \dot{x}_i^{(n)} \end{bmatrix} + \begin{bmatrix} 1-C \\ -S/T_c \end{bmatrix} p_x^*$$

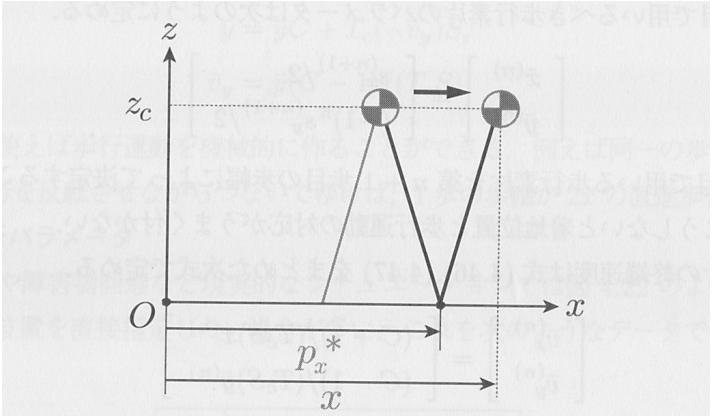


図 4.24　地面に設定した原点を基準に表現された線形倒立振子

3次元上での歩行パターン生成

- 着地点調整によるパターン生成

Step 1 歩行周期 T_{sup} および歩幅データ s_x, s_y を設定する．初期重心位置 (x, y) および，初期着地位置 $(p_x^*, p_y^*) = (p_x^{(0)}, p_y^{(0)})$ を設定する．

Step 2 $T := 0, n := 0$.

Step 3 時刻 T から $T + T_{sup}$ まで線形倒立振子の方程式 (4.51)（および y 軸用の式）を積分する．

Step 4 $T := T + T_{sup}, \quad n := n + 1$

Step 5 次の着地点 $(p_x^{(n)}, p_y^{(n)})$ を式 (4.48) により計算する．

Step 6 次の歩行素片 $(\bar{x}^{(n)}, \bar{y}^{(n)})$ を式 (4.49),(4.50) により計算する．

Step 7 目標状態 $(x^d, \dot{x}^d)$ を式 (4.55) により計算する．y 軸用の目標状態 $(y^d, \dot{y}^d)$ も対応する式で計算する．

Step 8 修正した着地位置 (p_x^*, p_y^*) を式 (4.57)(および y 軸用の式) により計算する．

Step 9 Step 3 へ戻る．

図 4.25　3 次元線形倒立振子に基づく歩行パターン生成アルゴリズム

3次元上での歩行パターン生成

- 着地点調整によるパターン生成
  - 直進歩行

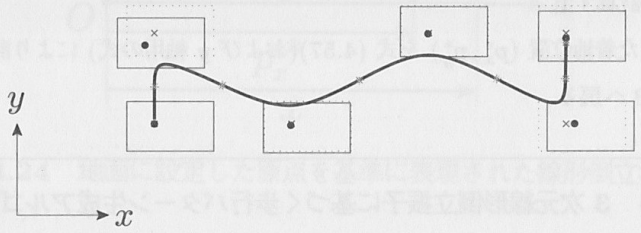


図 4.26　設計されたアルゴリズムにより作られた歩行パターン．太い曲線が重心軌道で，黒い点が着地点．目標着地点は × と破線で示されている．歩行開始と終了時には加減速を得るために設定からずれた着地点が選択されていることがわかる．$z_c = 0.8, T_{sup} = 0.8$，評価関数の重み $a = 10, b = 1$

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.3	0.3	0.3	0
$s_y^{(n)}$	0.2	0.2	0.2	0.2	0.2

3次元上での歩行パターン生成

- 着地点調整によるパターン生成
  - 斜め歩行

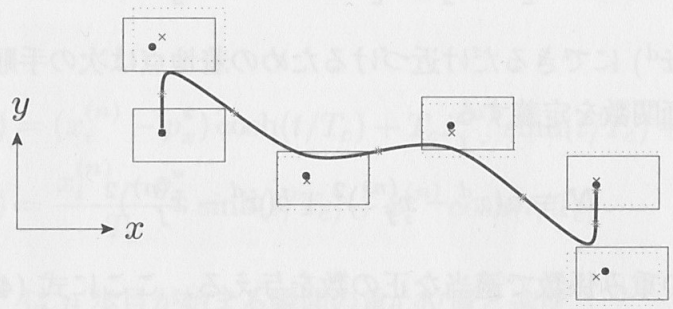


図 4.27　斜め歩行の例．横方向の移動量を得るために一歩ごとに横方向の歩幅 s_y を変化させている．$z_c = 0.8, T_{sup} = 0.8$，評価関数の重み $a = 10, b = 1$

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.2	0.2	0.2	0
$s_y^{(n)}$	0.2	0.3	0.1	0.3	0.2

3次元上での歩行パターン生成

- 方向転換
  - 方向転換をするには歩行パラメータに進行方向を表すデータを追加する必要がある

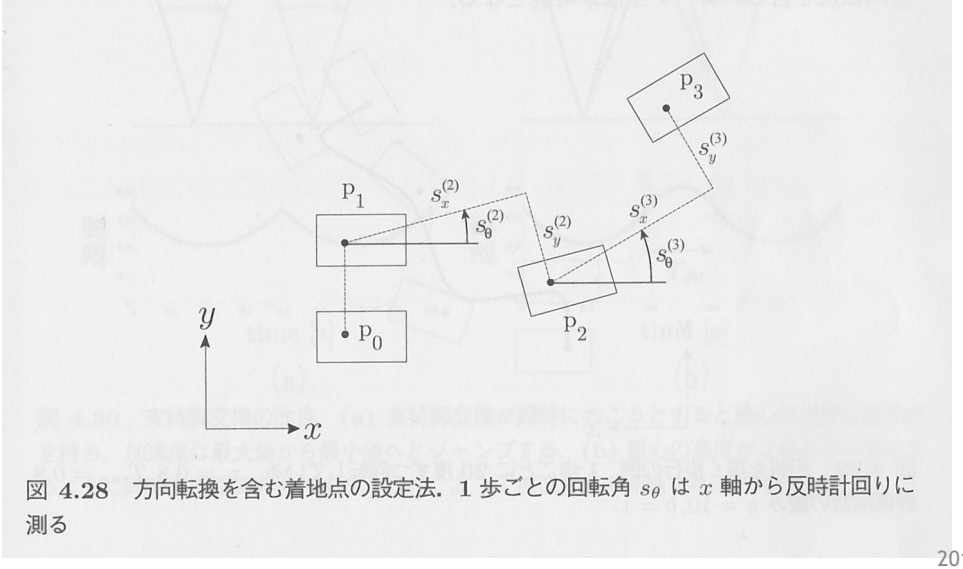


図 4.28　方向転換を含む着地点の設定法．1 歩ごとの回転角 s_θ は x 軸から反時計回りに測る

3次元上での歩行パターン生成

- 方向転換
- 回転角ｓθを加えた第ｎ歩目の着地位置（px, py）

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{bmatrix} + \begin{bmatrix} \cos s_\theta^{(n)} & -\sin s_\theta^{(n)} \\ \sin s_\theta^{(n)} & \cos s_\theta^{(n)} \end{bmatrix} \begin{bmatrix} s_x^{(n)} \\ -(-1)^n s_y^{(n)} \end{bmatrix}$$

- 回転角ｓθを加えた第ｎ歩目で用いるべき歩行素片のパラメータ（x，ｙ）

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

- 回転角ｓθを加えた歩行素片の終端速度(vx, vy)

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} (1+C)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix}$$

3次元上での歩行パターン生成

- 方向転換

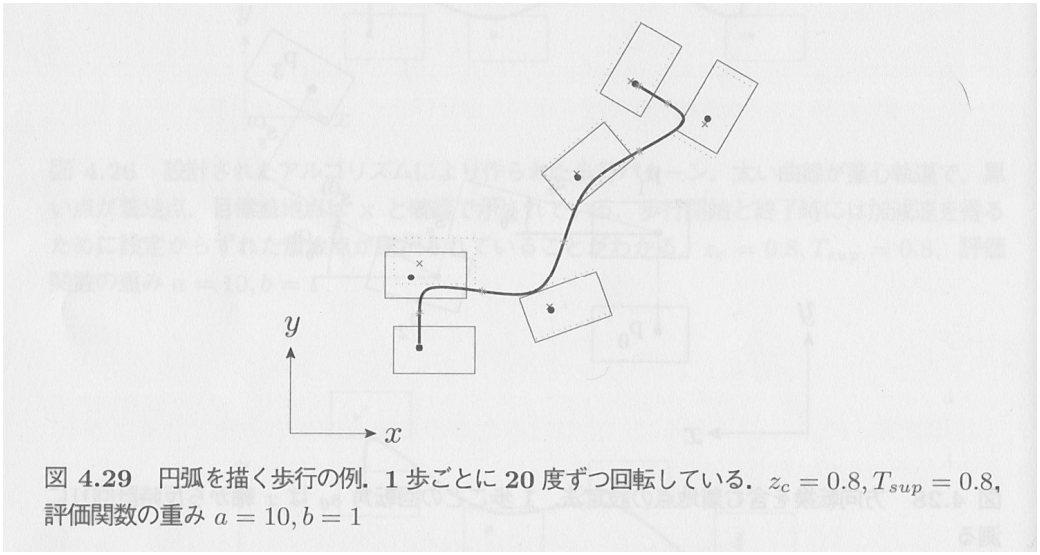


図 4.29　円弧を描く歩行の例．1 歩ごとに 20 度ずつ回転している．$z_c = 0.8, T_{sup} = 0.8$, 評価関数の重み $a = 10, b = 1$

n	1	2	3	4	5
s_x	0.0	0.25	0.25	0.25	0
s_y	0.2	0.2	0.2	0.2	0.2
s_θ	0	20	40	60	60

両足支持期間の導入

両足支持期間の導入

- これまで設計してきた歩行パターンでは、倒立振子の支持脚交換瞬時に行われていると仮定している。

- ロボットに大きな衝撃をもたらし、ダメージを与えてしまう

- それを防ぐには両足支持期間の導入を

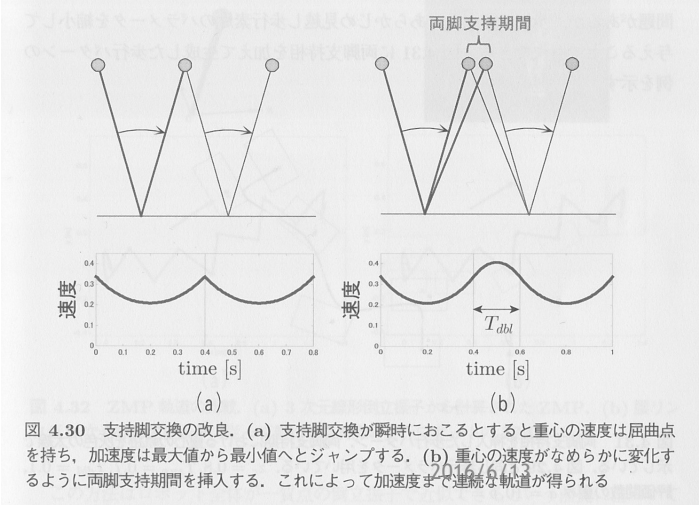


図 4.30　支持脚交換の改良．(a) 支持脚交換が瞬時におこるとすると重心の速度は屈曲点を持ち，加速度は最大値から最小値へとジャンプする．(b) 重心の速度がなめらかに変化するように両脚支持期間を挿入する．これによって加速度まで連続な軌道が得られる

両足支持期間の導入

- 下の式を使うと、右下の図のように導入できる

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \qquad (4.61)$$

係数 $a_0 \ldots a_4$ は支持脚交換時における状態（位置，速度，加速度）をもとに決定する．

- このような区間を設けると線形倒立振子で計画していたよりも歩幅が増大してしまう問題が

- 歩幅の増大を予め見越して計画する

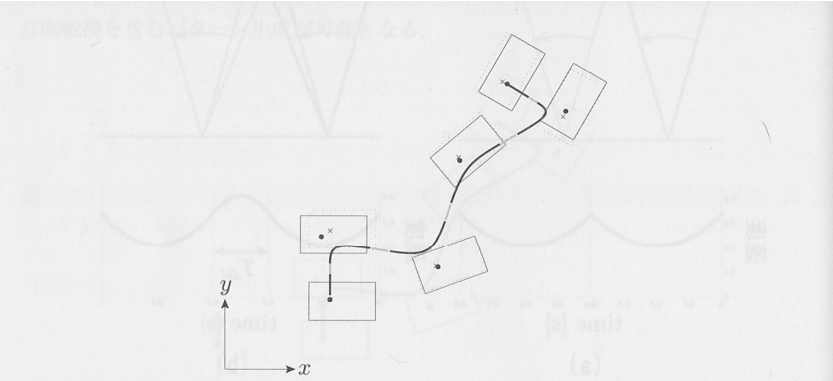


図 4.31　両脚支持相を挿入した歩行パターン．両脚支持期における重心の軌跡を灰色の太線で示している．図 4.29 と同じ歩行パラメータを用いている．$z_c = 0.8, T_{sup} = 0.7, T_{dbl} = 0.1$，評価関数の重み $a = 10, b = 1$

両足支持期間の導入

- 両足支持期間の導入で起きる問題
  - 導入は支持脚相の加速度をスムーズにしている一方で、遊脚の復帰に取れるまでの時間を短くしてしまう

  - 両脚支持期間Tdblをあまり大きくすることができない

最後に

- 今回の歩行パターンは、障害物がない平面上を歩行していることを前提にしている
  - つまり、凸凹道は、歩けない

- ZMPを規範とする歩行パターンを生成すればいい

引用元

- ヒューマノイドロボット（本）
  - http://www.amazon.co.jp/dp/4274200582
- ヒューマノイドロボットのフィードバック制御
  - http://www.topic.ad.jp/sice/papers/236/236-17.pdf